I claim:

1.   A method for software application development, the method comprising:

characterizing the file format and data structure of at least one known input file type; and

creating a library comprising a plurality of jobs, each job configured to perform a predetermined function and each job including an indicator of job termination, wherein the indicator is either termination success or termination failure, and the plurality of jobs are configured for linking according to the indicator, with at least one of the jobs configured to read the file format and data structure of the known input file and convert that input file to another file format or data structure.

2.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for extracting data from a file.

3.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for archiving files.

4.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for loading files.

5.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for messaging.

6.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for transforming data in a file.

7.   A method according to claim 1, wherein the library of jobs comprise at least one job configured for validating data in a file.

8.   A method according to claim 1, further comprising:

creating a job stream library, the job stream library comprising at least one job stream, the job stream comprising individual jobs from the library of jobs, and the at least one job stream in the job stream library including an indicator of job termination, wherein the indicator is either termination success or termination failure.

9.   A method according to claim 1, further comprising adding jobs to the library of jobs.

10.   A system for software application development, the system comprising:

means for characterizing the file format and data structure of at least one known input file type; and

means for creating a library comprising a plurality of jobs, each job configured to perform a predetermined function and each job including an indicator of job termination, wherein the indicator is either termination success or termination failure, and the plurality of jobs are configured for linking according to the indicator, with at least one of the jobs configured to read the file format and data structure of the known input file and convert that input file to another file format or data structure.

11.   Computer executable software code transmitted as an information signal, the code for software application development, the code comprising:

code to characterize the file format and data structure of at least one known input file type; and

code to create a library comprising a plurality of jobs, each job configured to perform a predetermined function and each job including an indicator of job termination, wherein the indicator is either termination success or termination failure, and the plurality

of jobs are configured for linking according to the indicator, with at least one of the jobs configured to read the file format and data structure of the known input file and convert that input file to another file format or data structure.

12.   A computer-readable medium having computer executable software code stored thereon, the code for software application development, the code comprising:

code to characterize the file format and data structure of at least one known input file type; and

code to create a library comprising a plurality of jobs, each job configured to perform a predetermined function and each job including an indicator of job termination, wherein the indicator is either termination success or termination failure, and the plurality of jobs are configured for linking according to the indicator, with at least one of the jobs configured to read the file format and data structure of the known input file and convert that input file to another file format or data structure.

13.   A programmed computer for software application development, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in the memory, wherein the program code comprises:

code to characterize the file format and data structure of at least one known input file type; and

code to create a library comprising a plurality of jobs, each job configured to perform a predetermined function and each job including an indicator of job

termination, wherein the indicator is either termination success or termination failure, and the plurality of jobs are configured for linking according to the indicator, with at least one of the jobs configured to read the file format and data structure of the known input file and convert that input file to another file format or data structure.

14. A method for processing a data file, the method comprising:

selecting a plurality of jobs from a library of jobs, each job including an indicator of job termination, wherein the indicator is either termination success or termination failure;

creating a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination;

creating a manager;

receiving the data file; and

processing the data file with the job stream using the manager.

15. A method according to claim 14, wherein receiving the data file comprises receiving the data file as an e-mail attachment.

16. A method according to claim 14, wherein receiving the data file comprises receiving the data file using file transfer protocol.

17. A method according to claim 14, wherein receiving the data file comprises receiving the data file using hypertext transfer protocol.

18. A method according to claim 14, wherein receiving the data file further comprises checking security access for the data file.

19. A method according to claim 14, further comprising creating a visual representation of the plurality of jobs and the interlinking of the jobs.

20. A method according to claim 14, further comprising creating an extensible markup language document corresponding to the job stream.

21. A system for software application development, comprising:

means for selecting a plurality of jobs from a library of jobs, each job including an indicator of job termination, wherein the indicator is either termination success or termination failure;

means for creating a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination;

means for creating a manager;

means for receiving the data file; and

means for processing the data file with the job stream using the manager.

22. Computer executable software code transmitted as an information signal, the code for software application development, the code comprising:

code to select a plurality of jobs from a library of jobs, each job including an indicator of job termination, wherein the indicator is either termination success or termination failure;

code to create a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination;

code to create a manager;

code to receive the data file; and

code to process the data file with the job stream using the manager.

23. A computer-readable medium having computer executable software code stored thereon, the code for software application development, the code comprising:

code to select a plurality of jobs from a library of jobs, each job including an indicator of job termination, wherein the indicator is either termination success or termination failure;

code to create a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination;

code to create a manager;

code to receive the data file; and

code to process the data file with the job stream using the manager.

24.   A programmed computer for software application development, comprising:

a memory having at least one region for storing computer executable program code; and

a processor for executing the program code stored in the memory, wherein the program code comprises:

code to select a plurality of jobs from a library of jobs, each job including an indicator of job termination, wherein the indicator is either termination success or termination failure;

code to create a job stream, the job stream comprising the plurality of jobs linked to each other according to the indicator of job termination;

code to create a manager;

code to receive the data file; and

code to process the data file with the job stream using the manager.